

[职业库](#) > [程序](#) > [游戏客户端](#) > AABB的FrustumCulling小记

## 原创 AABB的FrustumCulling小记

limli(李兵)

 发布时间:2022.08.31 16:54    15    0    0    [更多](#)
[分享至POPO眼界大开](#)

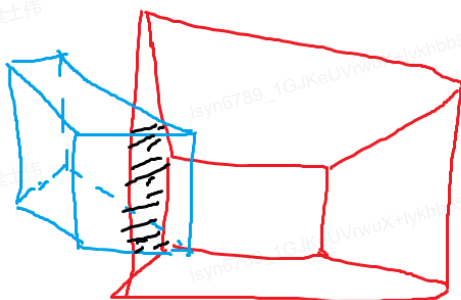
🔒 本文仅面向以下用户开放，请注意内容保密范围

查看权限: 互娱正式-默认推荐职业(程序-游戏;程序-运维&计费;程序-软件开发;QA;US;TA)

“ AABB的FrustumCulling一些方法的小结 ”

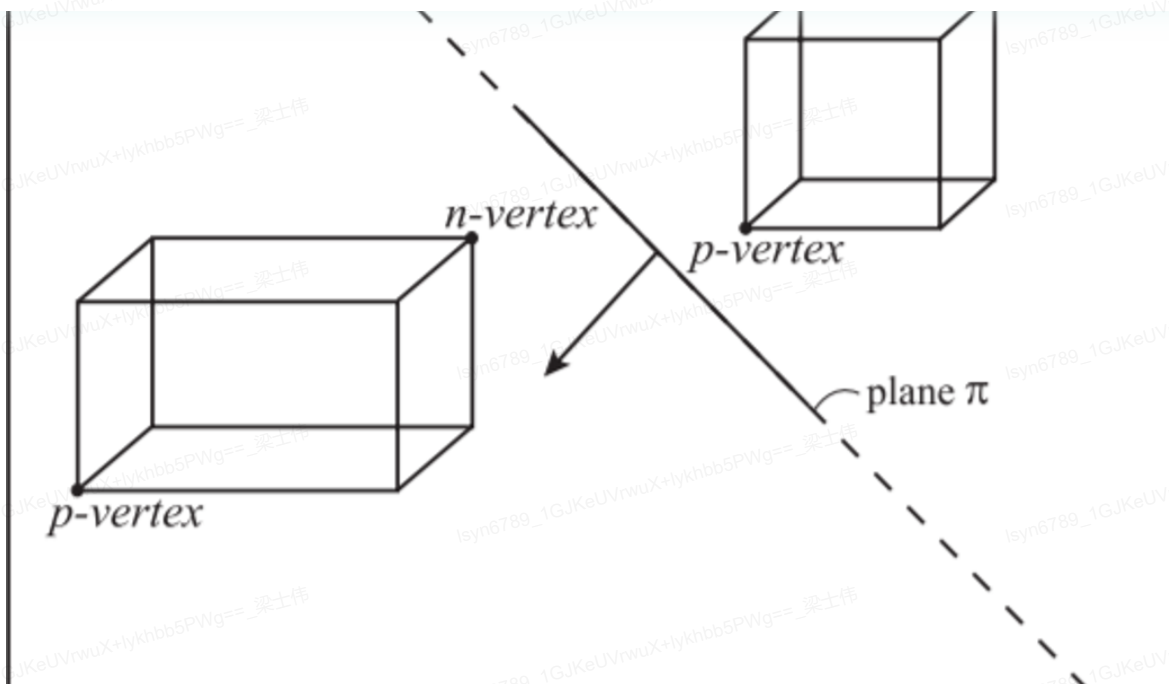
## AABB的FrustumCulling小记

最近需要实现一个FrustumCulling的相关工作，是基于AABB的，一看，这题我会，前不久刚看了个博客[Frustum Culling](#)，原理简单易懂，基本就是走一遍硬件的Frustum Culling的流程，把八个AABB的顶点转换到NDC中，判断是否在标准范围内（平台不同，Z的范围会不同），不亏是见多识广的我，搞定！等下，不对啊，好像即使八个顶点都在外面也可能与视锥相交啊，看我鼠标作画。



就像我画的那样（真生动形象），AABB可能穿过视锥。

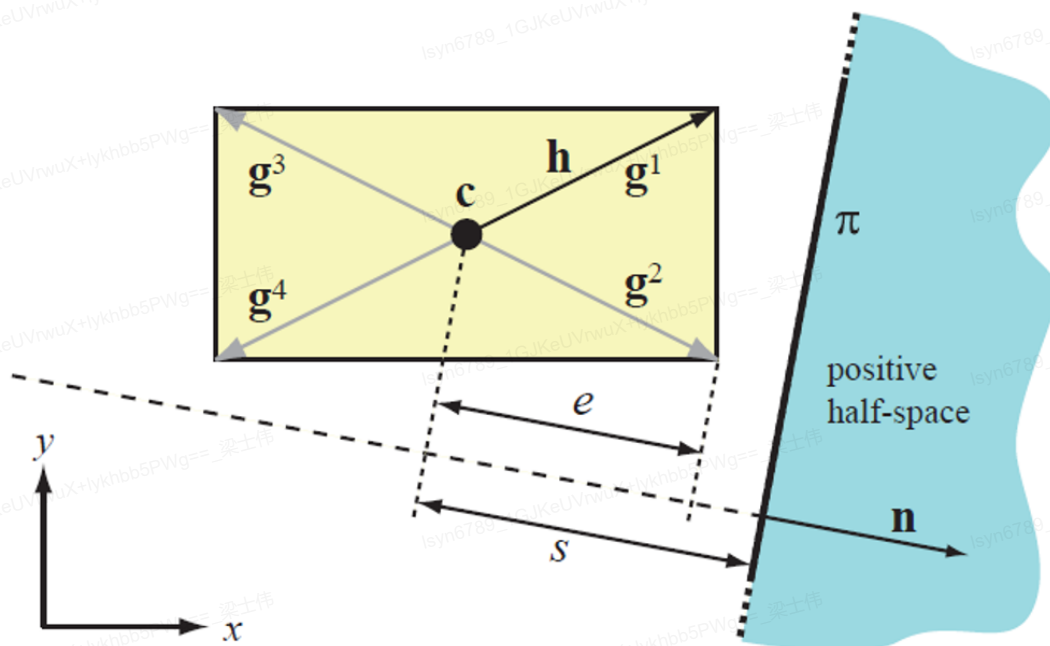
没事，我还会Google！找到了Milo Yip大大（念）的一个回答，里面提到了这篇论文[Optimized View Frustum Culling Algorithms for Bounding Boxes](#)，提供了一个保守剔除的方法，至少不会多剔。它是判断AABB的顶点是否在视锥体六个面的外面还是里面，它为了减少运算，找了离平面最小和最大的有向距离，也就是带符号的距离，当最大距离的点都在内部，就认为在内部，当最小距离都在外部，那就认为是在外部，否则的话就是相交。示意图如下：



其中n-vertex和p-vertex的计算方式是：

```
n-vertex = min(dot(box.p0, n), ... dot(box.p7, n))
p-vertex = max(dot(box.p0, n), ... dot(box.p7, n))
```

然后，我想是不是还有其他方法，这次就比较慎重的选择翻《Real-Time Rendering》，果然是还有更聪明的做法，《Real-Time Rendering》永远的神。



这个方法仍然是判断顶点和面的关系，主要的思路就是找到半对角向量在平面Normal上的最大投影，可以理解AABB在平面的Normal上的最大的Extent，然后拿这个值跟中心点到平面的距离作比较，就可以得到结果。其中e的计算也很巧妙，因为AABB会有八个方向的半角向量，也就是包括了三维坐标的所有象限，那最大的投影值一定是跟Normal同向的，所以可以直接拿第一象限的半角向量点乘上Normal的绝对值就可以了，伪代码如下：

```

returns({OUTSIDE, INSIDE, INTERSECTING});
1:  c = (bmax + bmin)/2
2:  h = (bmax - bmin)/2
3:  e = hx|nx + hy|ny + hz|nz
4:  s = c · n + d
5:  if(s - e > 0) return (OUTSIDE);
9:  if(s + e < 0) return (INSIDE);
10: return (INTERSECTING);
    
```

\*本内容仅代表个人观点，不代表网易游戏，仅供内部分享传播，不允许以任何形式外泄，否则追究法律责任。

☆ 收藏 0

👍 点赞 0

🔗 分享

📱 用手机查看



快来成为第一个打赏的人吧~

全部评论 0



请输入评论内容

还可以输入 500 个字



(可添加1个视频+5张图片)

☐ 匿名

评论

最热 最新



暂无评论

加载完毕,没有更多了

大家都在看



Share us  
your growing

常用链接

易协作

OA

会议

预定

文具

预定

游戏

部IT资

源

易网

网易

POPO

工作

报告

POPO服务  
号



KM APP  
下载